

# Laboratorio 4: Arquitectura y Organización de Computadores

**Profesor:** Viktor Tapia

**Ayudante de cátedra:** Mauricio Cortés

**Ayudante de Tarea:** Vicente Alvear y Luciano Yevenes

29 de Octubre 2023

## 1. Reglas Generales

Para esta tarea, debes utilizar el software QtARMSim<sup>1</sup> para escribir un programa en lenguaje de programación ARM Assembly que resuelva los problemas indicados en la sección de **enunciado**. Es importante destacar que el programa debe ser ejecutable en QtARMSim sin requerir la instalación de complementos adicionales o plugins.

## 2. Enunciado

Se le pide crear un programa en QtARMSIM que reciba como primer parámetro un número entero entre 1 y 2, y dependiendo de este número se ejecutara la “Función1” o la “Función2” respectivamente en ambas **cumpliendo la recursión de las funciones**.

### 2.1. Función1

El objetivo de la “funcion1” es llevar a cabo la traducción de un fragmento de código en Python, que se muestra en la **Figura 1**, a código ARM. La función deberá ser capaz de recibir dos variables enteras y mostrar el resultado por la consola de QtARMSIM.

```
1  def fuumo(x,n):
2      if n==0:
3          return 1.0 + 0.0*x
4      elif n==1:
5          return 2.0*x
6      else:
7          return 2.0*x*fuumo(x,n-1) - 2.0*(n-1)*fuumo(x,n-2)
```

*Figura 1: Función a reescribir*

---

<sup>1</sup> <https://pypi.org/project/qtarmsim/>

## 2.2. Función2

El número áureo es un número irracional que posee muchas propiedades interesantes, fue descubierto en la antigüedad y su valor numérico es  $\phi = \frac{1+\sqrt{5}}{2} = 1.618033 \dots$ . De las tantas formas que existen para calcularlo, podemos relacionar las conocidas sucesiones de Fibonacci y Lucas para determinar la potencia n-ésima  $\phi^n$  del número áureo de la siguiente forma:

$$\phi^n = \frac{L_n + F_n\sqrt{5}}{2}$$

Siendo  $F_n$  la n-ésima sucesión de Fibonacci y  $L_n$  la n-ésima sucesión de Lucas.

La función "funcion2" se le va a pedir calcular la n-ésima potencia aurea, para poder ayudarlos ya se ha escrito el código en Python como se muestra en la Figura 2, por lo que ustedes deben volver a traducir el código de Python a ARM. **Ya que QtARSim no maneja bien los puntos flotantes, se pide que solo imprima la parte entera del resultado de  $\phi^n$**

```
1  def Lucas(n):
2      if n == 0:
3          return 2
4      elif n == 1:
5          return 1
6      else:
7          return (Lucas(n - 1) + Lucas(n - 2))
8
9  def Fibonacci(n):
10     if n == 0:
11         return 0
12     elif n == 1:
13         return 1
14     else:
15         return (Fibonacci(n - 1) + Fibonacci(n - 2))
16
17  def aureo(n):
18     x = (Lucas(n) + Fibonacci(n)*(5**(1/2)))/2
19     return x
20
21  print(aureo(3))
22
```

Figura 2: Función para calcular en número áureo

Hint: Lucas y Fibonacci están relacionados por lo que  $F_n = \frac{L_{n-1} + L_{n+1}}{5}$

### 3. Entrada y salida de datos

La entrada y salida de datos se hará a través de las interfaces stdin (.data) y stdout (Pantalla LCD) de QtARMSim. Todos los valores numéricos estarán entre 0 y 25, incluyendo ambos límites.

- Entrada:

Valor	Función	Parámetros
1	Función1	Valores para x y n respectivamente
2	Función2	Valor para n

- Salida:



Figura 3: LCD donde deben aparecer los resultados

### 4. Datos ejemplo

Input	Output
1	4
2	
1	
1	14
2	
2	
1	34
3	
2	
2	1
0	
2	1
1	
2	2
2	
2	3
3	

## 5. README

Debe contener como mínimo:

- Nombre, Rol y Paralelo de los integrantes.
- Especificación de los algoritmos y desarrollo realizado.
- Supuestos utilizados

## 6. Consideraciones

- Se deberá trabajar de a pares. Se deberá entregar en Aula a más tardar el día 11 de Noviembre de 2023 a las 23:59 horas. Se descontarán 5 puntos por cada hora o fracción de atraso. Las copias serán evaluadas con nota 0 en el promedio de las tareas.
- La tarea debe realizarse usando el lenguaje ARM Assembly y el simulador QtARMSim.
- Pueden crear todas las funciones auxiliares que deseen, siempre y cuando estén debidamente comentadas.
- La entrega considera un único archivo de nombre t4.s junto con el README. Los archivos deberán ser comprimidos y enviados en un archivo.tar.gz o en .zip (esto queda a su conveniencia) con el nombre **LAB1\_ROL1\_ROL2**.
- Si no se entrega README, o si su programa no funciona, la nota es 0 hasta la corrección.
- Una vez entregadas las notas de la tarea existirá un plazo de 5 días para apelar. Transcurrido este plazo las notas no podrán ser modificadas.